

Dynamic Application Security Testing (DAST)

Key Points:

- Dynamic Application Security Testing (DAST) is a method used to find security vulnerabilities while an application is running. It's like probing an app from the outside, trying to break into it, much like a hacker would.
- **How it works:** Instead of just looking at the source code, DAST checks how the application behaves in real-time, searching for weak points or ways to exploit the system.

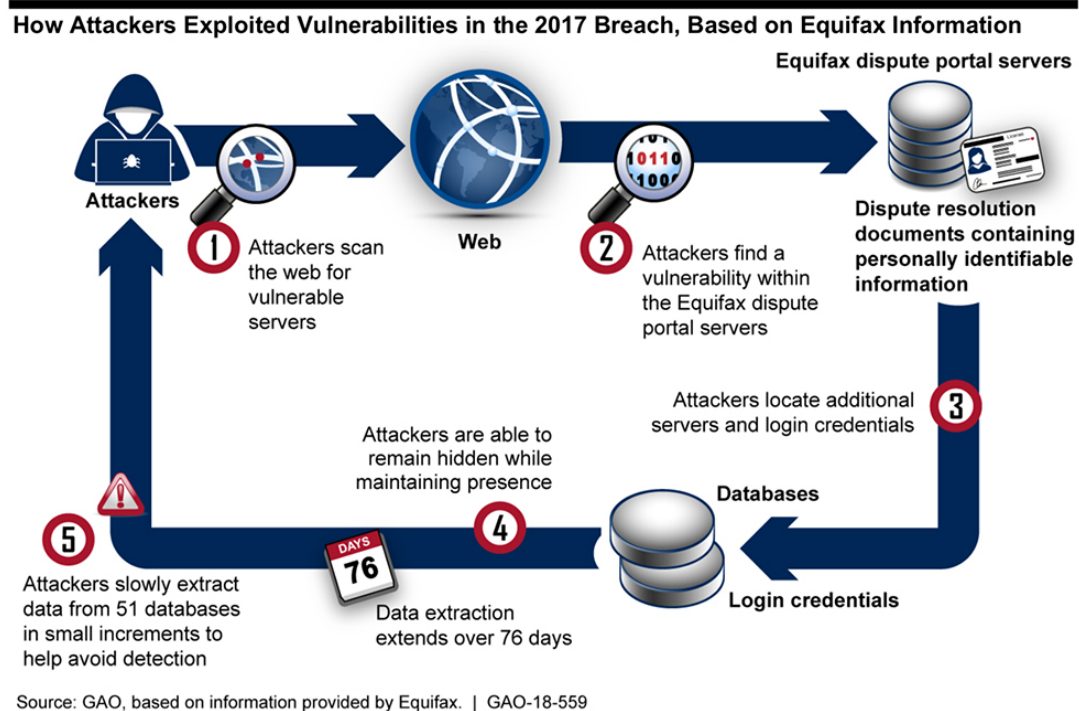


▼ Why Dynamic Testing Matters

- **Catches runtime issues:** Some problems only appear when the app is running, such as insecure configurations or weak authentication mechanisms. These vulnerabilities can't always be caught by simply looking at the code.
- **Real-world impact:** DAST helps protect against real-life attacks by finding weaknesses in your app before hackers do.

Use Case Example:

- **Equifax Data Breach (2017):** The attackers used a web vulnerability to execute malicious code, leading to the theft of personal data for millions of people. Had DAST been in place, it might have detected the issue in time to prevent the breach.



United States Government Accountability Office

<https://youtu.be/Y1BwWYVYS2E?si=ZPfeUc8MydBlEPSz>

▼ Common Web Vulnerabilities Detected by DAST

- **SQL Injection:** Attackers can manipulate database queries by injecting malicious inputs. DAST detects if your app is vulnerable to these injections.
- **Cross-Site Scripting (XSS):** This allows attackers to run malicious scripts on a user's browser. DAST simulates these attacks to check if your site is exposed.
- **CSRF:** Cross-Site Request Forgery forces users to perform actions they didn't intend. DAST helps spot these types of vulnerabilities.
- **Security Misconfigurations:** Incorrectly set security options, such as weak passwords or mismanaged permissions, can open your app to attacks.

▼ Real-World Examples of DAST in Action

GitHub's Security Practice:

- GitHub has integrated automated security testing, including DAST, into its workflow. By doing so, they are able to run vulnerability scans frequently and catch issues before they affect users.

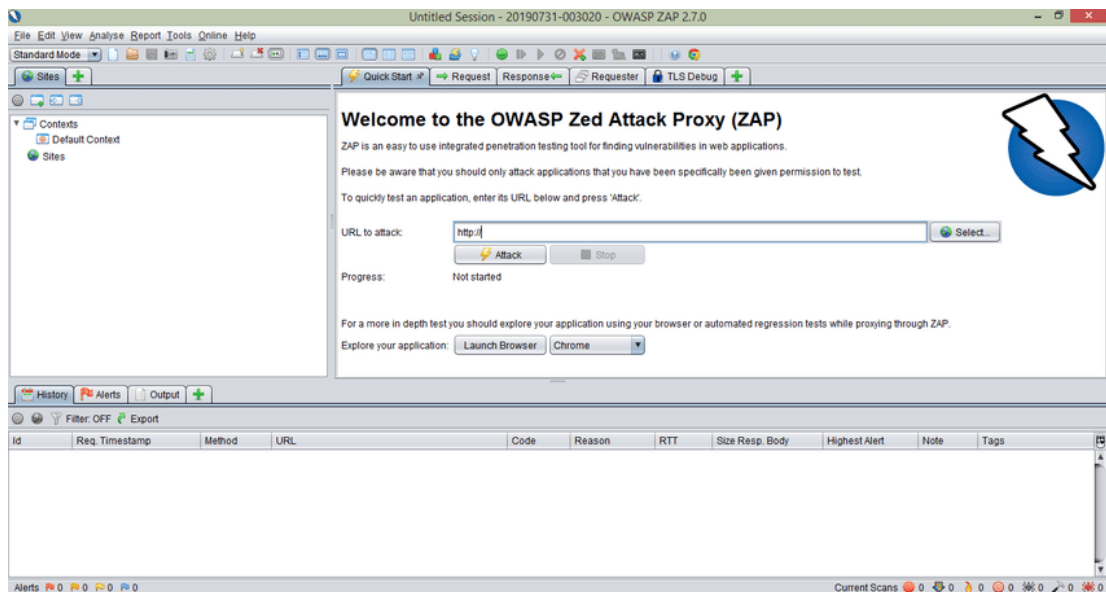
Capital One Data Breach (2019):

- The breach happened due to a misconfigured firewall that let hackers access sensitive data. A DAST tool might have detected this configuration flaw before it became a problem.

<https://www.youtube.com/watch?v=r7HV4s-4ksQ>

▼ OWASP ZAP – Tool Overview

- This is an open-source, easy-to-use tool for finding vulnerabilities in web apps while they run. It can perform scans, run attack simulations, and report on issues like SQL injection or XSS.





- **How it works:**

OWASP ZAP can act as a middleman between the web app and its users. It listens to the web traffic, spots vulnerabilities, and flags them.
- You can set up OWASP ZAP to automatically run scans during your CI/CD process (see below), ensuring that no vulnerable code gets deployed into production.

▼ Automating Dynamic Security Testing in Github Actions

ZAP Scan Baseline Report #93

 **Open** **github-actions** bot opened this issue 3 minutes ago · 0 comments



github-actions bot commented 3 minutes ago

- Site: <https://www.zaproxy.org>
New Alerts
 - **Strict-Transport-Security Header Not Set** [10035] total: 20:
 - <https://www.zaproxy.org/blog/2016-02-19-zap-newsletter-2016-february/images/image05.png>
 - <https://www.zaproxy.org/faq/index.xml>
 - <https://www.zaproxy.org/docs/desktop/addons/form-handler/images/formHandlerTable.PNG>
 - <https://www.zaproxy.org/docs/desktop/addons/hud/index.xml>
 - <https://www.zaproxy.org/docs/desktop/addons/websockets/images/106.png>
 - ..
 - **Cross-Domain Misconfiguration** [10098] total: 20:
 - <https://www.zaproxy.org/img/faq/supportAddonVersion.png>
 - <https://www.zaproxy.org/docs/desktop/addons/websockets/images/105.png>

- By integrating OWASP ZAP with GitHub Actions, you can ensure that every time a developer pushes new code, it's automatically tested for security vulnerabilities.
- **Example GitHub Actions Workflow:**

```
name: OWASP ZAP Scan

on:
  pull_request:
    branches:
      - main

jobs:
  zap_scan:
    runs-on: ubuntu-latest
    steps:
      - name: ZAP Scan
        uses: zaproxy/action-full-scan@v0.11.0
        with:
          target: 'https://www.zaproxy.org/'
```

▼ Benefits of Dynamic Security Testing

- **Continuous security checks:** Since DAST is integrated into CI/CD pipelines, it provides constant, real-time monitoring for vulnerabilities as new code is deployed.
 - **Reduced risk:** Regularly running dynamic tests means you catch potential security issues before they make it to production, reducing the chance of a breach.
 - **Actionable insights:** DAST tools give developers detailed reports, making it easier to understand and fix vulnerabilities quickly.
-

▼ Best Practices for Implementing DAST

- **Start early:** Introduce DAST in the development process as soon as possible, so vulnerabilities can be caught while they're still easy to fix.
 - **Run tests regularly:** Make DAST a regular part of your CI/CD pipeline to ensure security is maintained across all updates.
 - **Combine with other testing:** Use DAST alongside static analysis (SAST) and manual code reviews for complete coverage.
 - **Review reports promptly:** Ensure that all flagged vulnerabilities are promptly reviewed and addressed by the development team.
-